

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів  
Кафедра систем управління літальних апаратів

## Лабораторна робота № 3

з дисципліни «Алгоритмізація та програмування»  
на тему «Реалізація алгоритмів з розгалуженням мовою C++»

XAI.301. 175. 318. 08 ЛР

Виконав студент гр. \_\_\_\_\_ 318

\_\_\_\_\_ Каріна ГЛЄБОВА  
(підпис, дата) (П.І.Б.)

Перевірів

\_\_\_\_\_ к.т.н., доц. Олена ГАВРИЛЕНКО  
(підпис, дата) (П.І.Б.)

## МЕТА РОБОТИ

Вивчити теоретичний матеріал щодо синтаксису у мові C++ і подання у вигляді UML діаграм активності алгоритмів з розгалуженням та реалізувати алгоритми з використанням інструкцій умовного переходу і вибору мовою C++ в середовищі Visual Studio. Також опанувати та відпрацювати навички структурування програми з функціями.

## ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання на алгоритми з розгалуженням.

If5. Дано три цілих числа. Знайти кількість додатних і кількість від’ємних чисел в початковому наборі.

Завдання 2. Дано координати точки на площині (x, y). Визначити, чи потрапляє точка в фігуру заданого кольору (або групу фігур) і вивести відповідне повідомлення.

Geom4. Дано координати точки на площині (x, y). Визначити, чи потрапляє точка в фігуру червоного кольору (або групу фігур) і вивести відповідне повідомлення.

На рис. 1 показана умова завдання Geom 4.

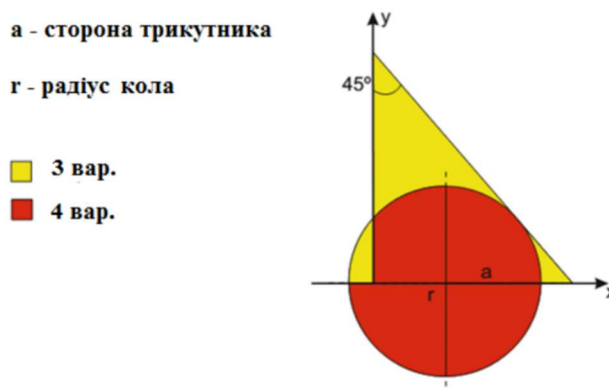


Рисунок 1 – Умова завдання Geom 4

Завдання 3. Обчислити площу і периметр плоскої фігури.

Geom4. Обчислити площу і периметр червоної фігури.

Завдання 4. Для вибору користувачем одного з трьох зазначених вище завдань розробити алгоритм організації меню в командному вікні з використанням інструкції вибору.

## ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі If 5.

Вхідні дані (ім'я, опис, тип, обмеження):

a – будь-яке ціле число.

b – будь-яке ціле число.

c – будь-яке ціле число.

Вихідні дані (ім'я, опис, тип):

positiveCount – кількість додатних чисел серед введених.

negativeCount – кількість від'ємних чисел серед введених.

На рис. 2 показаний приклад діаграми для завдання If 5.

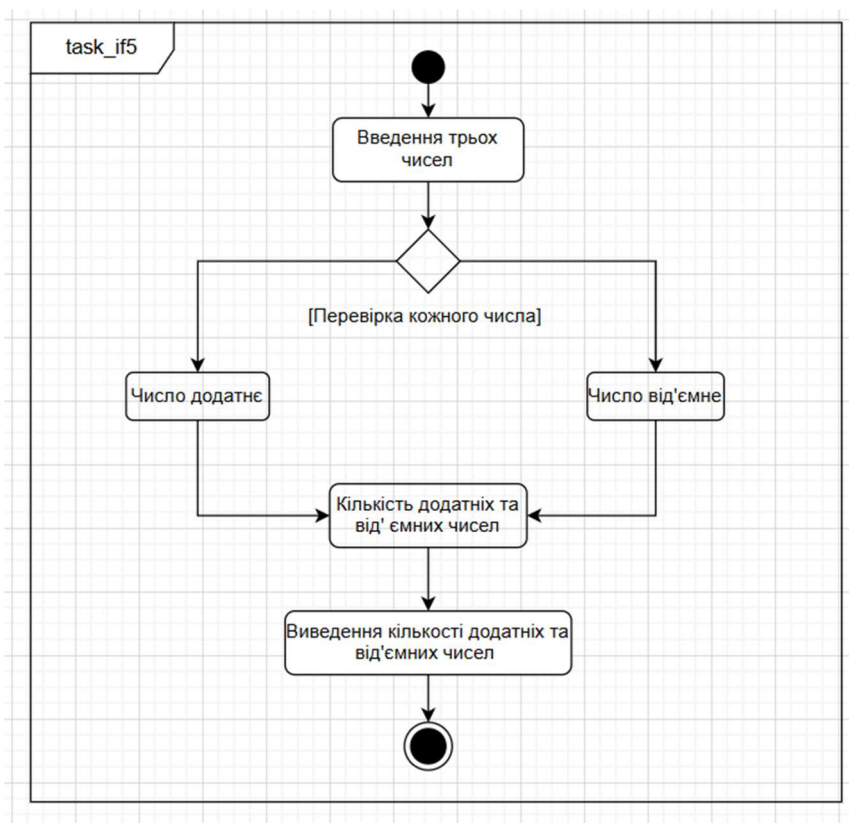


Рисунок 2 – Діаграма для завдання If 5

Лістинг коду вирішення задачі If 5 наведено в дод. А (стор. 8-9)  
Екран роботи програми показаний на рис. Б.1 (додат. Б, стор. 10)

Завдання 2, 3.

Вирішення задачі Geom 4.

Вхідні дані (ім'я, опис, тип, обмеження):

float x, y – дійсні числа, координати точки.

float r – дійсне число, радіус кола.

const double pi – дійсне число із подвійною точністю; число пі.

double theta\_deg – дійсне число із подвійною точністю, кут у градусах, утворений хордою кола.

Вихідні дані (ім'я, опис, тип):

float area\_circle – дійсне число, площа повного кола.

float cath\_small\_yell\_tris – дійсне число, четверта частина радіуса, відрізок жовтої частини на колі.

double zmensh – дійсне число із подвійною точністю, зменшуване у формулі знаходження повної площі хорди.

double vdmk – дійсне число із подвійною точністю, зменшуване у формулі знаходження повної площі хорди.

double S\_chord – дійсне число із подвійною точністю, площа повної хорди.

double area\_yellow – дійсне число із подвійною точністю, площа жовтої частини на колі.

float real\_circle – дійсне число, площа фігури без урахування жовтої частини кола.

double L\_chord – дійсне число із подвійною точністю, периметр фігури, утвореної хордою.

double theta\_rad – дійсне число із подвійною точністю; кут, утворений хордою, у радіанах.

double L\_arc – дійсне число із подвійною точністю, довжина дуги.

double P\_chord – дійсне число із подвійною точністю, периметр жовтої частини кола.

double perimeter – дійсне число із подвійною точністю, периметр червоної частини кола.

На рис. 3 показаний приклад діаграми для завдання Geom 4.

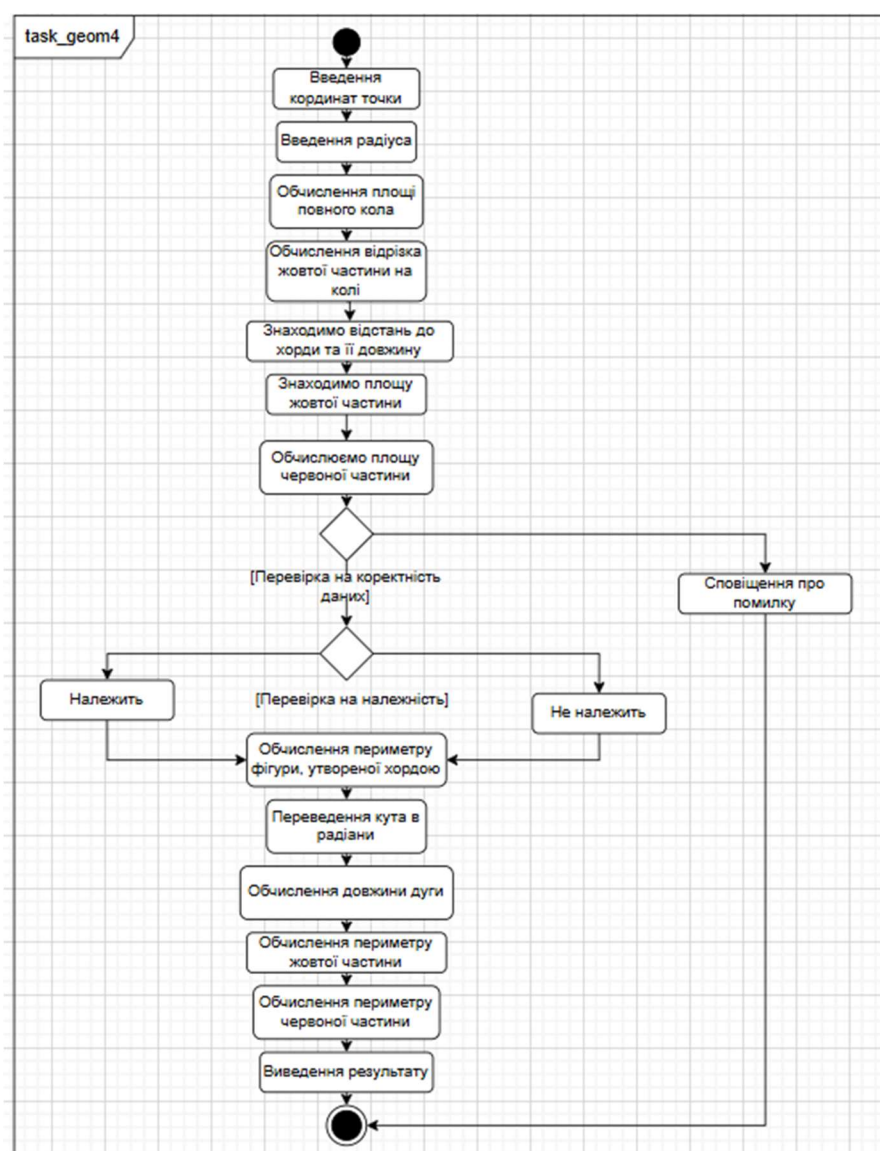


Рисунок 3 – Діаграма для завдання Geom 4

Лістинг коду вирішення задачі Geom 4 наведено в дод. А (стор. 8-9).  
Екран роботи програми показаний на рис. Б.2 (додат. Б, стор. 10)

Завдання 4.

Організація меню.

Вхідні дані (ім'я, опис, тип, обмеження):

“Номер завдання:” – введення номеру завдання.

Вихідні дані (імя, опис, тип):

task\_if5 – якщо ввели число «1».

task\_geom4 – якщо ввели число «2».

На рис. 4 показаний приклад діаграми для завдання 4

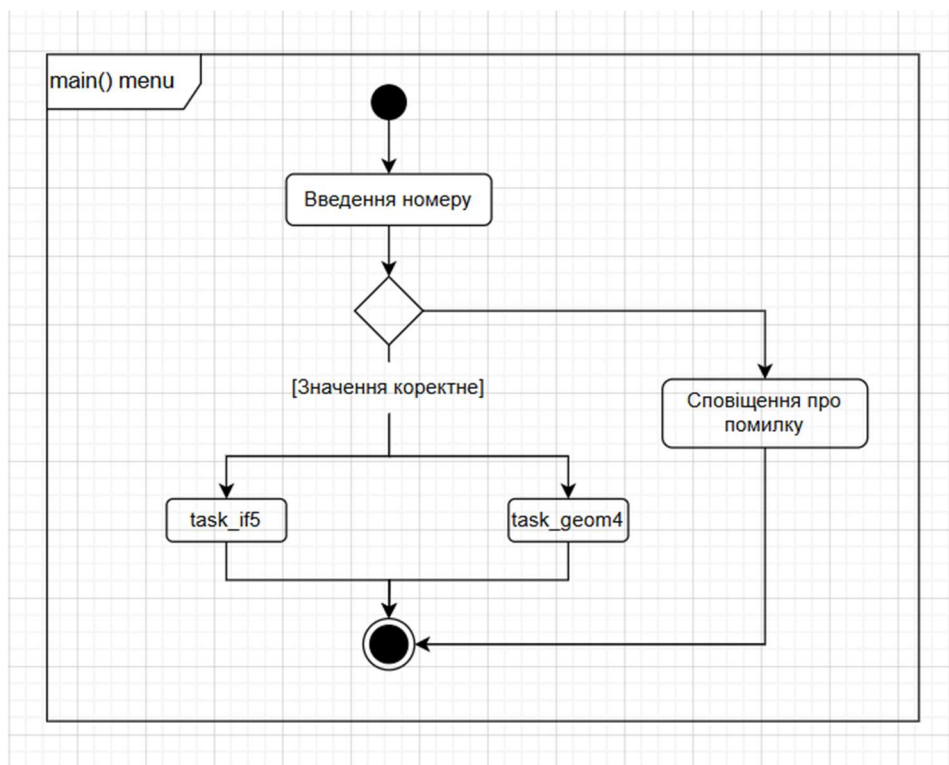


Рисунок 4 – Діаграма для завдання 4

## ВИСНОВКИ

Було вивчено синтаксис мови C++ і закріплено на практиці подання алгоритмів із розгалуженням у вигляді UML діаграм активності та відпрацьовано структурування програми з функціями і викликом меню. Було реалізовано алгоритми з використанням інструкцій умовного переходу і вибору мовою C++.

## ДОДАТОК А

### Лістинг коду програми

```

#include <iostream>
#include "windows.h"
#include <cmath>
using namespace std;

//If5.Дано три цілих числа. Знайти кількість додатних та від'ємних чисел в початковому
наборі
void task_if5(); //завдання 1, декларація функції

// Дано координати точки на площині (x, y).
// Визначити, чи потрапляє точка в фігуру заданого кольору (або групу фігур)
// і вивести відповідне повідомлення.
void task_geom4(); //завдання 2, декларація функції

int main()
{
    SetConsoleOutputCP(1251);
    int menu;
    cout << "Номер завдання:";
    cin >> menu;
    // перемикання між завданнями
    switch (menu)
    {
        case 1: task_if5(); break; // Завдання 1
        case 2: task_geom4(); break; // Завдання 2
        default: cout << "Помилка! (Тільки 1,2)" << endl; // Повідомлення про помилку
    }
    system("pause");
    return 0;
}

// Завдання 1 реалізація
void task_if5()
{
    int a, b, c;
    cout << "***** If 5 *****" << endl;
    cout << "Введіть три цілих числа:"; // введення чисел
    cin >> a >> b >> c;
    int positiveCount = 0, negativeCount = 0; // ініціалізація змінних

    if (a > 0) positiveCount++; // перевірка кожного числа окремо
    if (a < 0) negativeCount++;
    if (b > 0) positiveCount++;
    if (b < 0) negativeCount++;
    if (c > 0) positiveCount++;
    if (c < 0) negativeCount++;

    cout << "Кількість додатніх чисел:" << positiveCount << endl; // виведення
результату
    cout << "Кількість від'ємних чисел:" << negativeCount << endl;
}

// Завдання 2, реалізація
void task_geom4()
{
    float x, y, r;
    const double pi = 3.141592;
    double theta_deg = 90; //центральний кут у градусах, утворений хордою

```



```

cout << "***** Point in geometry region 4 *****" << endl;
cout << "Введіть координати точки:"; //введення координат точки
cin >> x >> y;
cout << "Введіть радіус кола:"; //введення радіуса кола
cin >> r;

float area_circle = pi * r * r; //обчислення площі кола
float cath_small_yell_tris = r / 4; //знаходимо відрізок жовтої частини на колі
float dist_2_chord = r - cath_small_yell_tris; //обчислюємо відстань до хорди

//знаходимо площу повної хорди (розкладаємо на зменшуваче та від'ємник)
double zmesh = r * r * acos(dist_2_chord / r);
double vdmk = dist_2_chord * sqrt(r * r - dist_2_chord * dist_2_chord);

double S_chord = zmesh - vdmk;

double area_yellow = S_chord / 2; //знаходимо площу жовтої частини на колі
float real_circle = area_circle - area_yellow; //знаходимо площу червоної
частини кола

if (r <= 0) //перевірка коректності даних
    cout << "Повинно бути додатнім!";
else {
    if (x >= 0 && y >= 0 && (x * x + y * y <= r * r)) {
        //перевірка на належність точки до 1-ї чверті кола
        cout << "Точка належить до 1-ї чверті." << endl; //виведення
результатів
    }
    else if ((x <= 0 && y >= 0 && (x * x + y * y <= r * r)) && (x * x + y * y
<= area_yellow)) {
        //перевірка на належність до 2-ї чверті
        cout << "Точка належить до 2-ї чверті." << endl; //виведення
результатів
    }
    else if (x <= 0 && y <= 0 && (x * x + y * y <= r * r)) {
        //перевірка на належність до 3-ї чверті
        cout << "Точка належить до 3-ї чверті." << endl; //виведення
результатів
    }
    else if (x >= 0 && y <= 0 && (x * x + y * y <= r * r)) {
        //перевірка на належність до 4-ї чверті кола
        cout << "Точка належить до 4-ї чверті." << endl; //виведення
результатів
    }
    else {
        cout << "Точка не належить колу." << endl;
    }
}

double L_chord = 2 * sqrt(r * r - dist_2_chord * dist_2_chord); //обчислення
периметру фігури, яку утворює хорда
double theta_rad = theta_deg * pi / 180; //переведення кута в радіани
double L_arc = r * theta_rad; //обчислення довжини дуги
double P_chord = (L_chord + L_arc) / 2; //обчислення периметру жовтої частини
кола (фігури, яку утворює хорда)

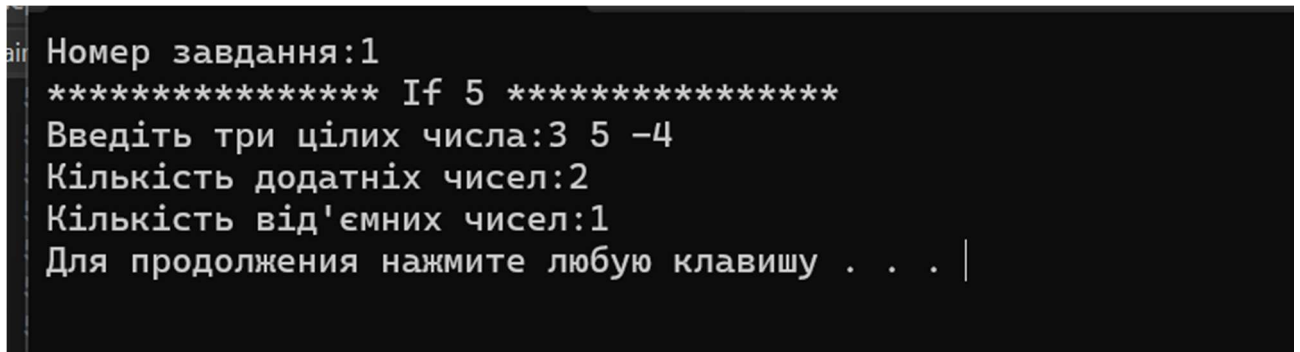
double perimeter = 2 * pi * r - P_chord; //обчислення периметру червоної частини
кола

cout << "Площа фігури:" << real_circle << endl; //виведення результату
cout << "Периметер фігури:" << perimeter << endl;
}

```

ДОДАТОК Б  
Скрін-шоти вікна виконання програми

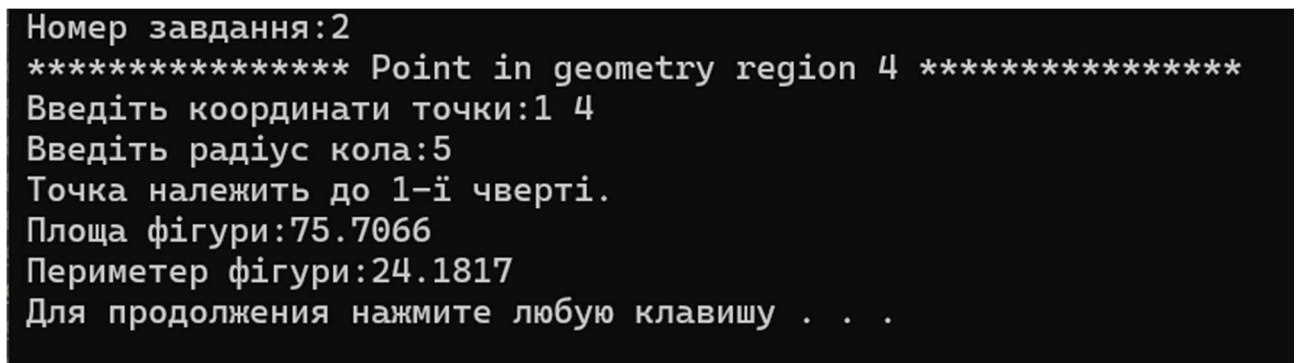
На рис. Б.1 показаний приклад виконання завдання 1.



```
Номер завдання:1
***** If 5 *****
Введіть три цілих числа:3 5 -4
Кількість додатніх чисел:2
Кількість від'ємних чисел:1
Для продовження натисніть будь-яку клавішу . . . |
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання 1

На рис. Б.2 показаний приклад виконання завдання 2.



```
Номер завдання:2
***** Point in geometry region 4 *****
Введіть координати точки:1 4
Введіть радіус кола:5
Точка належить до 1-ї чверті.
Площа фігури:75.7066
Периметер фігури:24.1817
Для продовження натисніть будь-яку клавішу . . .
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання 2